Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 7, n. 1, 2020.

Trabalho apresentado no XXXIX CNMAC, Uberlândia - MG, 2019.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

---

Estudo da Eficiência das Acelerações de Convergência do Método da Potência

Hedjany Sena da Silva[1]
Modesto Valci Moreira Lopes[2]
Ivan Mezzomo[3]
Matheus da Silva Menezes[4]
Departamento Ciências Naturais, Matemática e Estatística, UFERSA, Mossoró, RN
Manoel Benedito Neto[5]
Escola de Ciência e Tecnologia, UFRN, Natal, RN

1 Referencial Teórico

Este trabalho examina o Método da Potência (MP) e duas modificações com aceleração de convergência: O Método da Potência com Aceleração de Aitken (MPA) e o Método da Potência com Deslocamento (MPD). Esses métodos estimam o autovalor dominante de uma matriz real sem haver a necessidade de realizar o cálculo do polinômio característico. O objetivo desse trabalho é investigar se essas acelerações se mostram mais eficientes para os problemas em análise tanto em relação ao número de iterações quanto em relação ao tempo de processamento. O MP e o MPD são dados pelo teorema 1 e 2, respectivamente.

Teorema 1: [2] *Dado uma matriz real quadrada A de ordem n e seus autovalores $\lambda_1, \lambda_2, \ldots, \lambda_n$ com seus correspondentes autovetores $u_1, u_2, \ldots, u_n$. Suponha que os autovetores são lineramente independentes e que $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Seja a sequência y_k definida por $y_{k+1} = Ay_k$, com $k = 1, 2, \ldots,$ onde y_0 é um vetor arbitrário que permite a expansão $y_0 = \sum_{j=1}^{n} c_j u_j$, com c_j escalares quaisquer e $c_1 \neq 0$, então $\lim_{k\to\infty} \dfrac{(y_{k+1})_r}{(y_k)_r} = \lambda_1$, onde r indica a r-ésima componente.*

Teorema 2: [2] *Dado uma matriz real quadrada A de ordem n e seus autovalores $\lambda_1, \lambda_2, \ldots, \lambda_n$ com seus correspondentes autovetores $u_1, u_2, \ldots, u_n$. Suponha que os autovetores são lineramente independentes e que $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Seja a sequência y_k definida por $x_{k+1} = (A - qI)x_k$ onde I é a matriz identidade, q é um parâmetro qualquer e x_0 é um vetor arbitrário que permite a expansão: $x_0 = \sum_{j=1}^{n} \beta_j v_j$, com β_j escalares quaisquer e $\beta_1 \neq 0$. Então, $\lim_{k\to\infty} \dfrac{(x_{k+1})_r}{(x_k)_r} = \lambda_1$.*

O Método de Aitken Δ^2 pode ser utilizado para acelerar a convergência de uma sequência linearmente convergente, conforme teorema abaixo.

---

[1]hedjany@icloud.com
[2]modsval@gmail.com
[3]imezzomo@ufersa.edu.br
[4]matheus@ufersa.edu.br
[5]manoelneto_1@hotmail.com

Teorema 3: [1] *Suponha que* $\{P_n\}_{n=0}^{\infty}$ *é uma sequência linearmente convergente com limite* P*. Para motivarmos a criação de uma sequência* $\{P'_n\}_{n=0}^{\infty}$ *que converge mais rapidamente para* P *do que* $\{P_n\}_{n=0}^{\infty}$*, primeiro assumimos que as expressões* $P_n - P$*,* $P_{n+1} - P$ *e* $P_{n+2} - P$ *coincidam e* n *é suficientemente grande tal que* $P = P_n - \frac{(P_{n+1}-P_n)^2}{P_{n+2}-2P_{n+1}+P_n}$.

2 Resultados e Discussões

Todas as matrizes são simétricas e a maior parte delas são diagonal dominante e definida positiva. Visando analisar a funcionalidade do algoritmo proposto, efetuamos a implementação em software SciLab 5.5.2 em máquina com processador intel i5, 4GB de RAM e sistema operacional Windows 10. Como critério de parada, usamos o teste do erro absoluto para cada componente de λ_1 dada por $|\lambda_1^{k+1} - \lambda_1^{k}|_r < \varepsilon$, com precisão $\varepsilon < 10^{-5}$.

Tabela 1: Resultado dos experimentos realizados

Problema	Método	Autovalor	# Iteração	Eficiência	Tempo	Eficiência
	MP	6197287.060444	239		1.335	
bcsstk05	MPA	6197287.646445	168	+29.71%	1.143	+14.38%
(Ord 153)	MPD	6197287.188500	309	-29.29%	1.556	-16.55%
	MP	4359568,057780	167		0.928	
fidap005	MPA	4359568,050222	123	+26.35%	0.978	-5.39%
(Ord 27)	MPD	4359567,910585	205	-22.75%	0.922	+0.65%
	MP	11,534504	22		1.007	
gr_ 30_ 30	MPA	11,843866	75	- 240.90%	1.236	-22.74%
(Ord 900)	MPD	11,470014	210	- 854.54%	5.042	-400.70%
	MP	66,447392	81		1.444	
sherman4	MPA	66,496697	141	- 71.43%	1.722	-19.25%
(Ord 1104)	MPD	65,981088	227	- 180.25%	7.322	- 407.06%
	MP	3015178995,408649	809		2.105	
bcsstk01	MPA	3015178995,476635	250	+ 69.10%	1.363	+35.25%
(Ord 48)	MPD	3015178995,450425	772	+ 4.57%	1.688	+ 19.81%
	MP	1474,857449	3		0.744	
1138_bus	MPA	1474,857467	5	- 66.67%	0.731	+1.84%
(Ord 1138)	MPD	1475,110240	16	- 433.33%	1.225	- 64.65%

A partir da análise dos dados da Tabela 1, podemos notar que em relação ao número de iterações, o MPD foi menos eficiente em relação ao MP em até 854,54%, exceto para a matriz bcssrk01, que apresentou eficiência de 4,57%. Por sua vez, o MPA se mostrou mais eficiente em 3 matrizes em até 69,10% em relação ao MP e menos eficiente em outras 3 matrizes até 240,90% em relação ao MP. Em relação ao tempo de processamento, o MPD foi menos eficiente em até 407,06% em quatro problemas e se mostrou mais eficiente nos problemas fidap005 (0,65%) e bcsstk01 (19,81%). Já o MPA foi menos eficiente em até 22,74% em relação ao MP nos problemas fidap005, gr_30_30 e sherman4, e apresentou eficiência em até 35,25% nos problemas bcsstk05, bcsstk01 e1138_bus.

Agradecimentos: Os autores agradecem o apoio da UFERSA e do CNPq.

Referências

[1] R. L. Burden, D. Faires and A. M. Burden. *Análise Numérica, 3a edição.* Cengage Learning, São Paulo, 2015.

[2] N. B. Franco. *Cálculo Numérico, 6a Edição.* Pearson, São Paulo, 2006.

[3] MatrixMarket, disponível em https://math.nist.gov/MatrixMarket/. Acesso em: 26/04/2019.

[4] Florida Sparse Matrix Collection, disponível em: https://sparse.tamu.edu. Acesso em: 26/04/2019.

